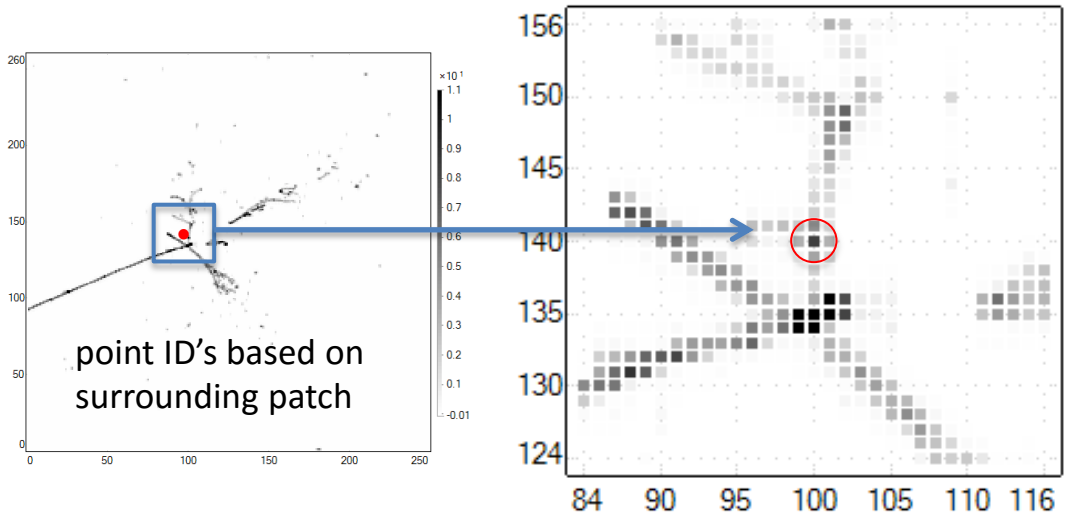
CNN-based tools development

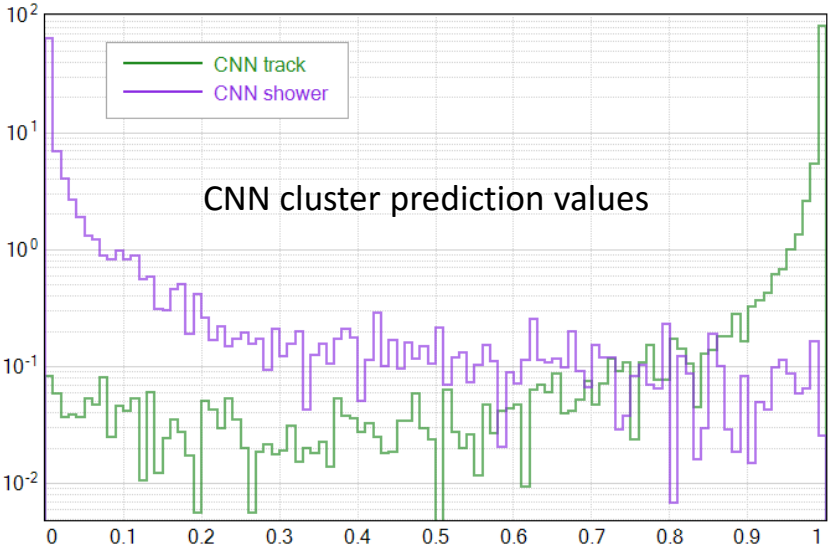
P. Płoński, D. Stefan, R. Sulej and seems others are joining

- basic idea, first application: EM vs. track-like ID
- implementations in / outside LArSoft
- list of tasks for the next weeks, and longer term

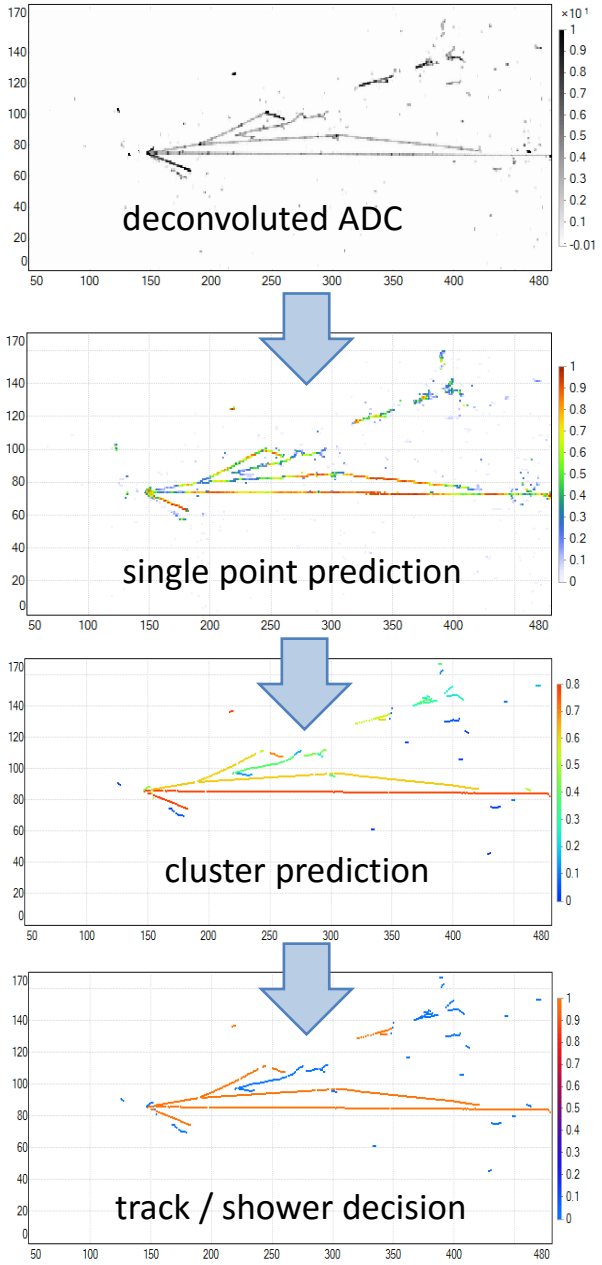
Basic idea: support standard reco, now used for **EM-like / track-like cluster identification:**

(as usual, old CNN on these pictures!)



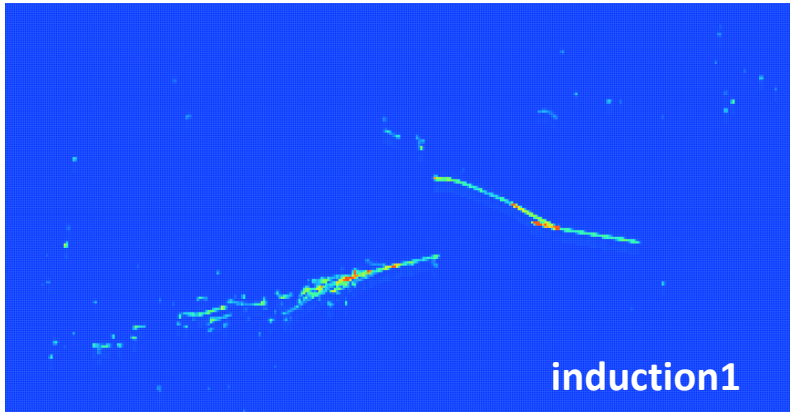


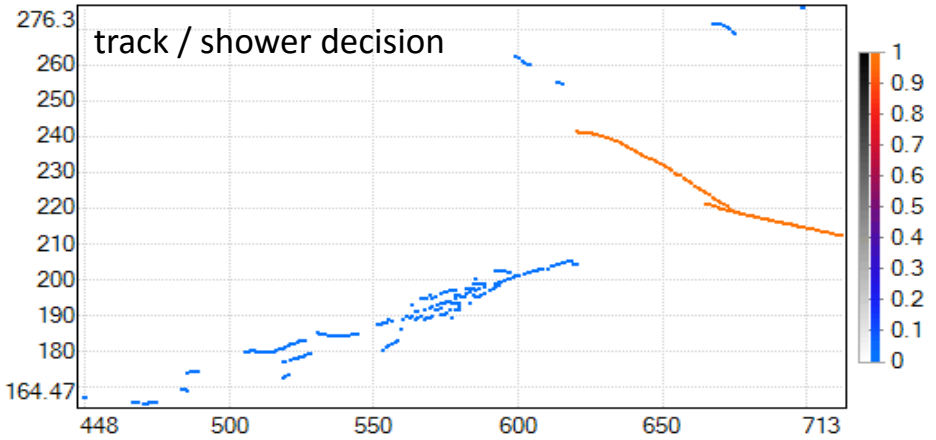
EM-like / track-like flow:



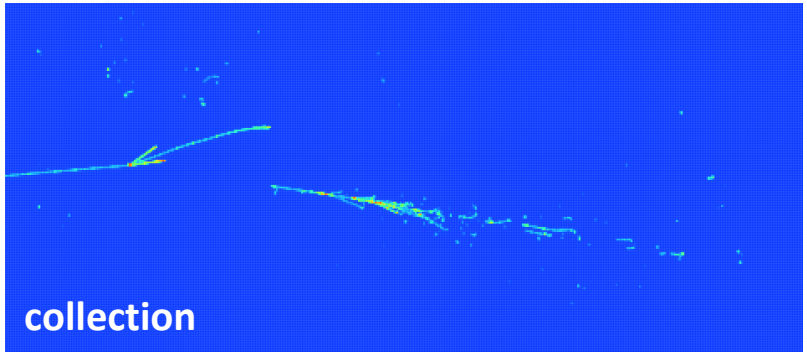
CNN results as of today: π^+ 2 GeV/c in protoDUNE SP

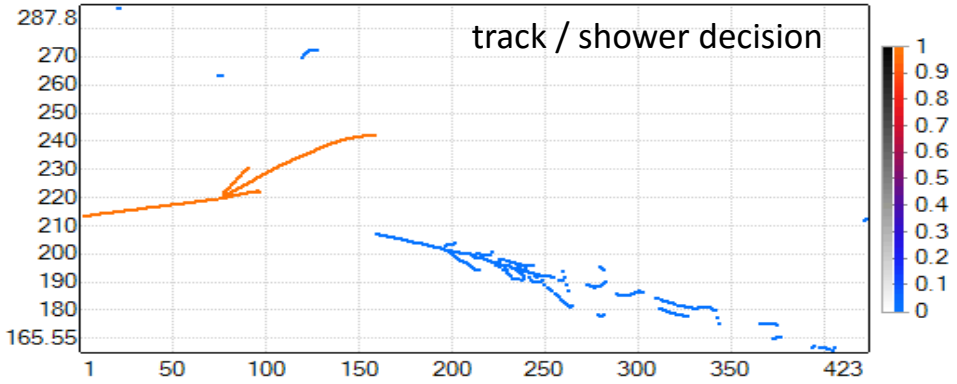
96.2% track / **96.6%** EM correct cluster ID rate (2GeV/c π^+ in protoDUNE)





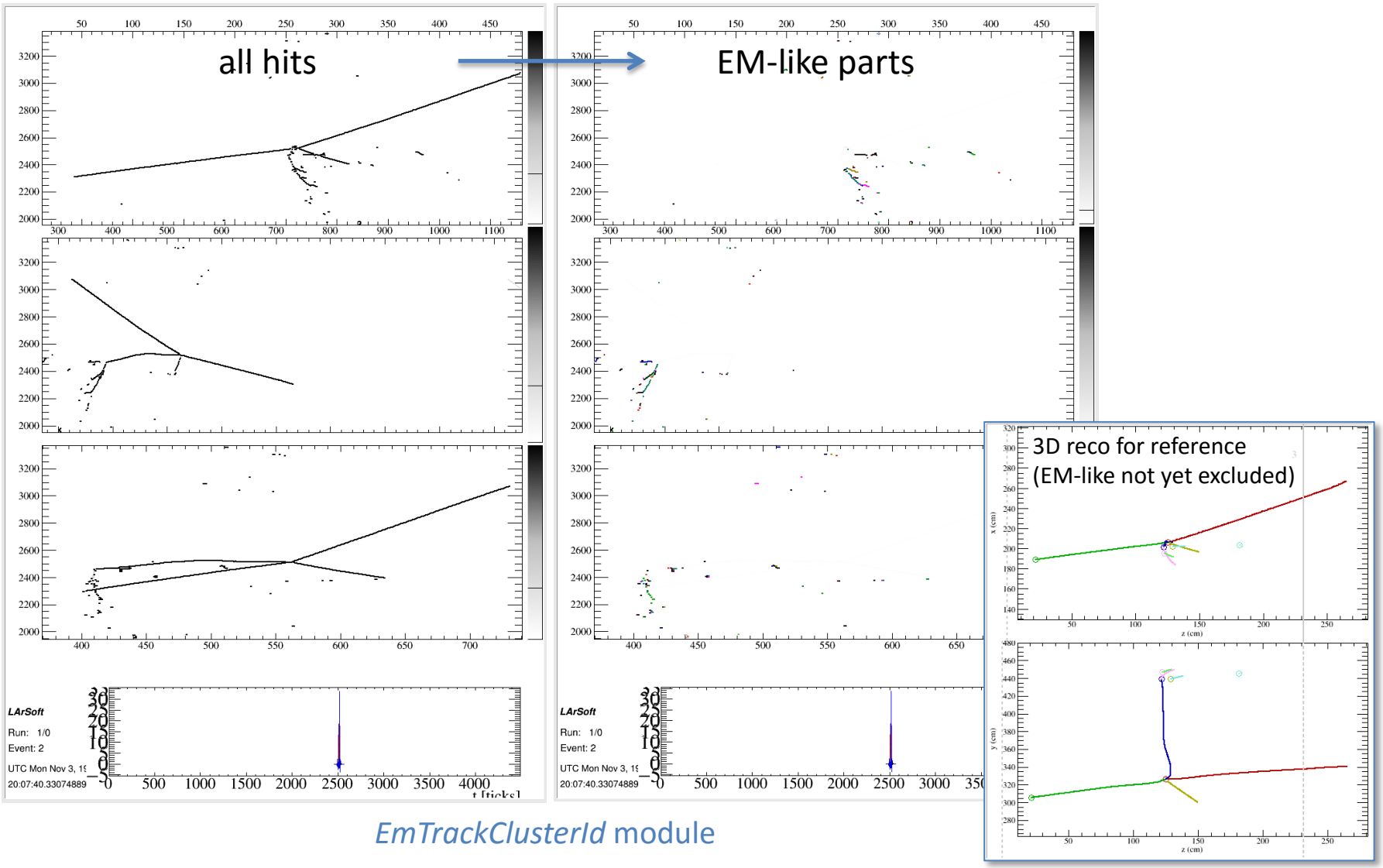
All OK





CNN results as of today: π^+ 2 GeV/c in protoDUNE SP

- module for tagging clusters and unclustered hits pushed to develop



EmTrackClusterId module

EM/track ID: data sizes…

- now used patch 32x32 (DUNE, 5mm pitch, downscale x10, needed square patch) = 16cm x 19cm
  - need large enough context and level of details
  - need similar size in wires and drift
  - need enough topologies for training (~5-8M patches / ~10k events was OK)

  ➔ **O(10GB) training set** (and larger files in the preparation stages…)

- Larger patch and drift resolution has more information for sure!
  - but GPU memory limitations are close…
  - and more border effects for larger patches

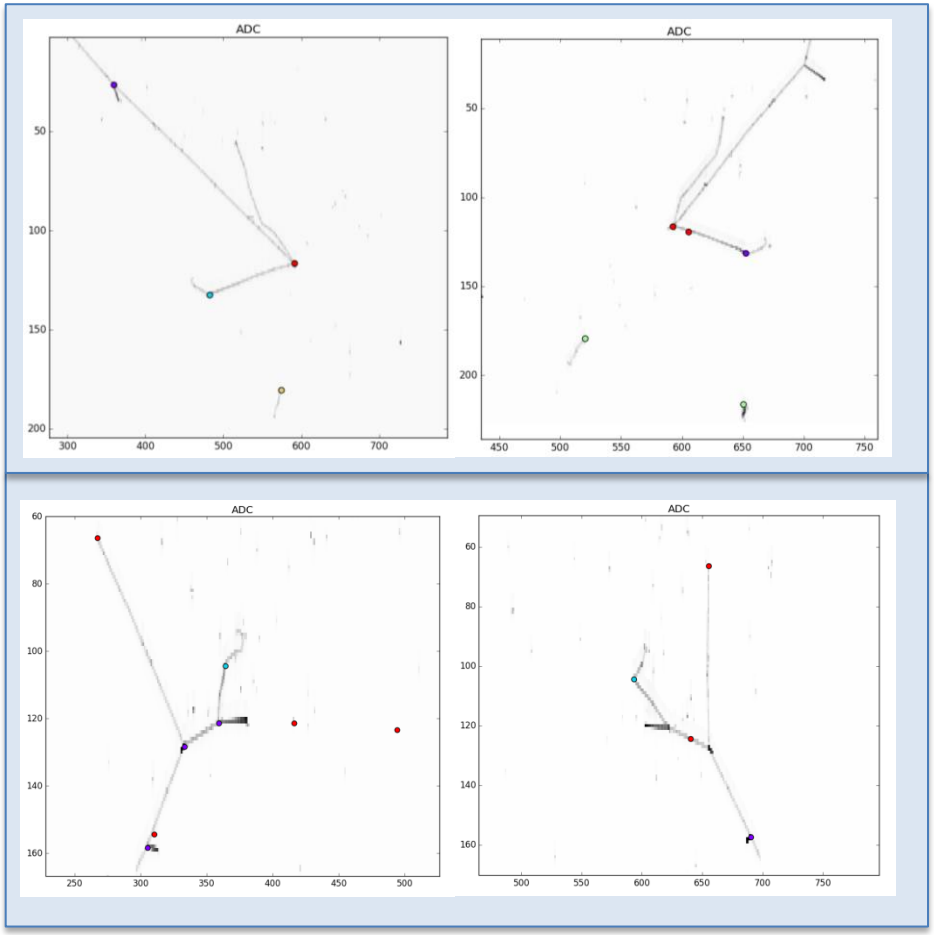
- rectangular patches are on the way (larreco feature/rnd_HigherDriftResCNN)
  - includes also downsampling other than max-pool

- 4mm, area 16cm^2: patch 40x40, downscale x6

  → will start from these parameters, already a bit more drift resolution

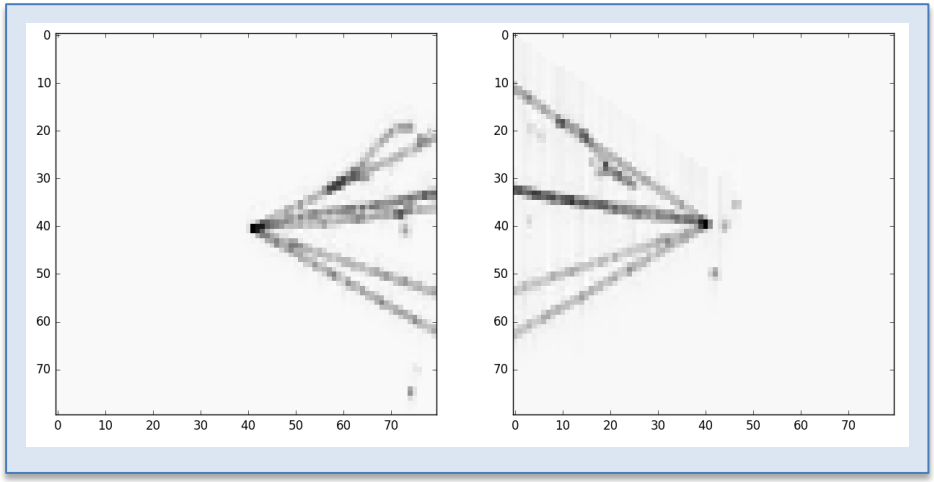
Vertex classification: similar „basic block” to EM/track ID

- Vertex identification
  - support tracking with **interaction/decay finding**
  - **select EM shower starting points** (not trivial in low energy)



data preparation module ~ready, need to verify thresholds to be tuned for reasonable visibility criteria

- Neutrino interaction classification
  - force classifier to be focused on the vertex features
  - **try to be sensitive to the „gap”** in full neutrino events



- need more events to build training set (only 1 training image pair/triplet per 1 event)
- more complex (than very simple) CNN may be needed
- uses larger patch arount the vertex and less downsampled drift
- more careful when producing data files to avoid really huge volumes…

How to approach work for the next weeks (LArIAT, protoDUNE, DUNE ν's – very close)

- EM/track ID: need „LArIAT" model (= 4mm pitch)
- Vertex / decay point ID / classification: data prep needs last tuning of MC truth use

- Prepare (downscaled) wire dumps from MC **(in LArSoft)**
  - these files are *our starting point* to make all variations of training sets for EM/track and Vtx ID tasks → carefully
  - they are downsampled ROI's: small enough to move around the world
  - when needed: prepare next sets with different downsampling method / scale

- Make training / testing **(no LArSoft, Python scripts: big big data sets, GPU training)**
  - first set: patch size / downsample method and scale; make first CNN
  - more sets: play with statistics, training target definition; make useful/not overtrained CNN

- Test **(back in LArSoft)**
  - systematic checks: noise, amplitude impact, MC/data, …

- Massive automatic search for optimal architecture / training method

few times

many times

More work…

- EM/track ID
- Vertex / decay point ID / classification

Dorota, Jessica, Robert

- Neutrino vertex classification

Dorota (need another approach to make data, and DUNE work…)

- Systematic effects

…likely will become a big task, start training yourself now

- Model / trainig optimization

Piotr and his MLJAR framework

- Data product design (vector of PID values, +some description, to be assigned to hits & clusters)

Other approaches: longer term

- RNN for localization of e.g. decaying μ
- Combine patch-size CNN with larger context
- Tensorflow interface fo LArSoft